МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине

‘Информационные системы и базы данных’

Вариант №336760

Выполнил:
Студент группы P33312
Соболев Иван
Александрович
Преподаватель:
Николаев Владимир
Вячеславович

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2023

Задание:

Лабораторная работа #1

Для выполнения лабораторной работы №1 необходимо:

1. На основе предложенной предметной области (текста) составить ее описание. Из полученного описания выделить сущности, их атрибуты и связи.
2. Составить инфологическую модель.
3. Составить даталогическую модель. При описании типов данных для атрибутов должны использоваться типы из СУБД PostgreSQL.
4. Реализовать даталогическую модель в PostgreSQL. При описании и реализации даталогической модели должны учитываться ограничения целостности, которые характерны для полученной предметной области.
5. Заполнить созданные таблицы тестовыми данными.

Для создания объектов базы данных у каждого студента есть своя схема. Название схемы соответствует имени пользователя в базе studs (sXXXXXX). Команда для подключения к базе studs:

psql -h pg -d studs

Каждый студент должен использовать свою схему при работе над лабораторной работой №1 (а также в рамках выполнения 2, 3 и 4 этапа курсовой работы).

Текст варианта:

Диаспара нет ничего, кроме пустыни. Пожалуйста, отведите его туда, если можете. Кто знает, вдруг вам известен путь наружу... Когда он столкнется с реальностью, это, наверное, позволит излечить некоторые странности его сознания...

Описание предметной области:

По приведенному тексту и остальному произведению, откуда взят данный текст, можно составить следующую предметную область:

Есть мир после апокалипсиса, в нем живут люди и есть города, некоторые города разрушены. Люди живут в этих городах, кто-то путешествует между ними, кто-то охраняет склады с оружием, провизией или лекарствами.

Существуют **города**, некоторые из которых разрушены, то есть город имеет статус разрушения – разрушен/не разрушен. Также существуют **люди**, у которых есть имена и профессия. Каждый человек имеет одно **сознание** (1:1), у сознания могут быть некоторые **странности**. В одном сознании может быть несколько странностей (1:M). В каждом городе **находятся** некоторые **склады**, склады могут находиться в нескольких городах, при этом в городе может находиться несколько складов. Каждый склад может содержать либо **оружие**, либо **еду**, либо **лекарства** (1:M). Каждый склад может содержать несколько складов, при этом один артефакт хранится только на одном складе. Также люди могут передвигаться в города, для этого существуют **пути** (M:M). Каждый человек может выбрать несколько путей передвижения, по одному пути могут идти несколько человек. В город можно прийти несколькими путями, при этом путь ведет только в один город.

Список сущностей и их классификация:

Стержневые:

- Город
  - cityId – уникальный идентификатор города.
  - cityName – название города.

    - destructionStatus – статус разрушения.
    - coordinateX – координата X.
    - coordinateY – координата Y.
- Человек
    - humanId – уникальный идентификатор человека.
    - humanName – имя человека.
    - profession – профессия человека.
- Склад
    - storageId – уникальный идентификатор склада.
    - storageName – название склада.
    - capacity – вместимость склада (в $м^3$).
- Сеть складов
    - networkId – уникальный идентификатор сети.
    - networkName – название сети.
    - reputation – репутация сети от 0 до 100.
- Странности
    - odditiesId – уникальный идентификатор странностей.
    - odditiesType – тип странностей.
    - description – описание странностей.

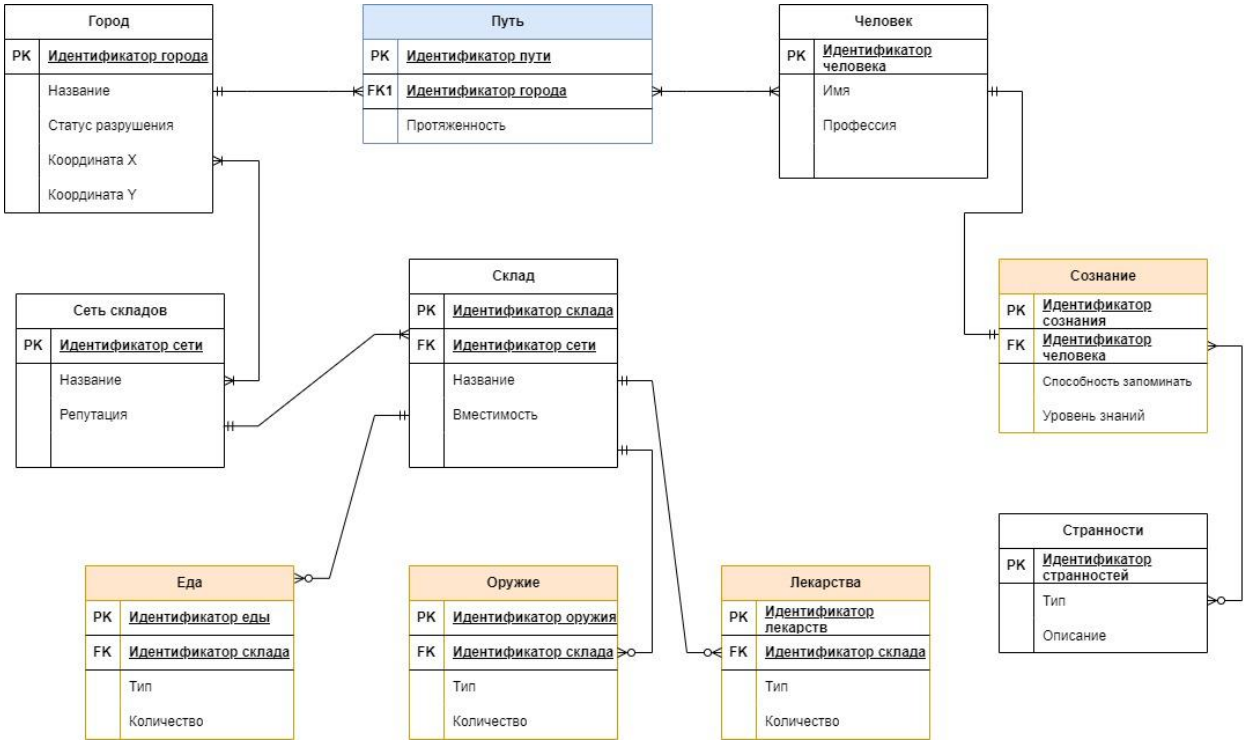
Ассоциативные:

- Путь
    - pathId – идентификатор пути.
    - cityId – идентификатор города.
    - length – протяженность пути.

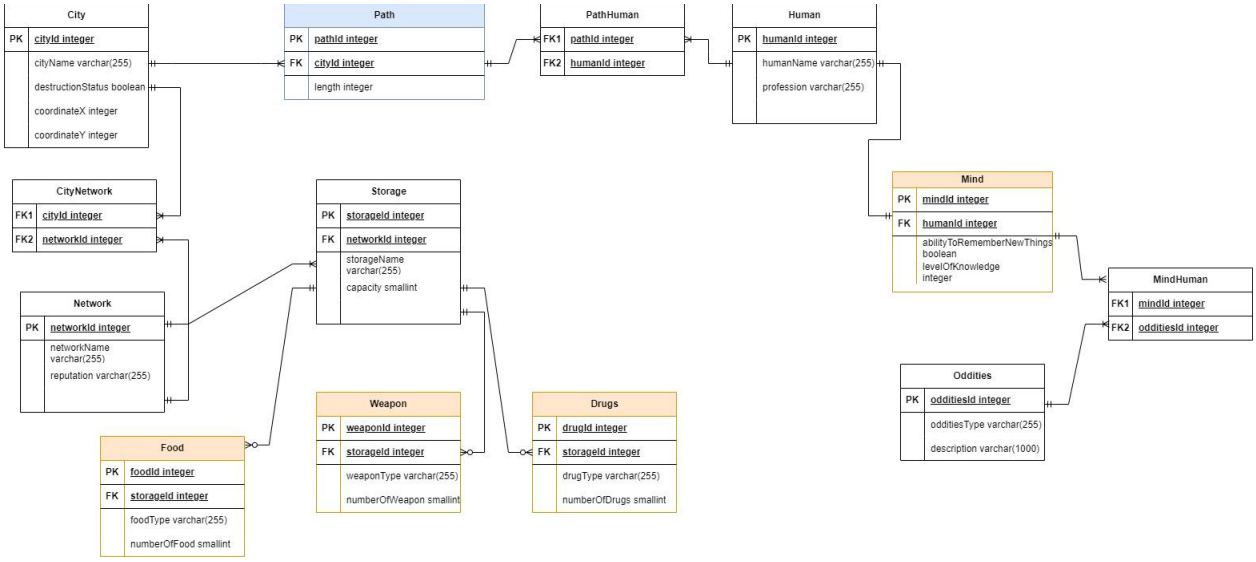
Характеристические:

- Оружие
    - weaponId – уникальный идентификатор оружия.
    - weaponType – тип оружия.
    - numberOfWeapon – количество оружия.
    - storageId – идентификатор склада, на котором хранится оружие.
- Еда
    - foodId – уникальный идентификатор еды.
    - foodType – тип еды.
    - numberOfFood – количество еды.
    - storageId – идентификатор склада, на котором хранится еда.
- Лекарства
    - drugId – уникальный идентификатор лекарства.
    - drugType – тип лекарств.
    - numberOfDrugs – количество лекарств.
    - storageId – идентификатор склада, на котором хранятся лекарства.
- Сознание
    - mindId – уникальный идентификатор сознания.
    - humanId – идентификатор человека, которому принадлежит.
    - abilityToRememberNewThings – способность запоминать новые данные
    - levelOfKnowledge – уровень знаний в сознании (IQ)

Инфологическая модель:



Даталогическая модель:



Реализация даталогической модели на SQL:

Создание таблиц:

```
CREATE TABLE IF NOT EXISTS City
(
    cityId            SERIAL PRIMARY KEY,
    cityName          VARCHAR(255) NOT NULL,
    destructionStatus BOOLEAN DEFAULT false,
    coordinateX       Integer      NOT NULL,
    coordinateY       Integer      NOT NULL
);

CREATE TABLE IF NOT EXISTS Human
(
    humanId    SERIAL PRIMARY KEY,
    humanName  VARCHAR(255) NOT NULL,
    profession VARCHAR(255) NOT NULL
);

CREATE TABLE IF NOT EXISTS Network
(
    networkId   SERIAL PRIMARY KEY,
    networkName VARCHAR(255) NOT NULL,
    reputation  INTEGER      NOT NULL
        CHECK (reputation >= 0 and reputation <= 100)
);

CREATE TABLE IF NOT EXISTS Storage
(
    storageId   SERIAL PRIMARY KEY,
    networkId   INTEGER REFERENCES Network ON DELETE CASCADE NOT NULL,
    storageName VARCHAR(255)                                 NOT NULL,
    capacity    smallint                                     NOT NULL
        CHECK (capacity > 0)
);

CREATE TABLE IF NOT EXISTS Path
(
    pathId  SERIAL PRIMARY KEY,
    cityId  INTEGER REFERENCES City ON DELETE CASCADE  NOT NULL,
    humanId INTEGER REFERENCES Human ON DELETE CASCADE NOT NULL,
    length  INTEGER                                    NOT NULL
        CHECK (length > 0)
);



CREATE TABLE IF NOT EXISTS CityNetwork
(
    cityId    INTEGER NOT NULL,
    networkId INTEGER NOT NULL,
    CONSTRAINT cityId_fk FOREIGN KEY (cityId) REFERENCES City (cityId) ON
DELETE CASCADE,
    CONSTRAINT networkId_fk FOREIGN KEY (networkId) REFERENCES Network
(networkId) ON DELETE CASCADE,
    CONSTRAINT cityNetworkId PRIMARY KEY (
                                          cityId, networkId
        )
);



CREATE TABLE IF NOT EXISTS Mind
```

```
(
    mindId                    SERIAL PRIMARY KEY,
    humanId                   INTEGER REFERENCES Human ON DELETE CASCADE NOT
NULL,
    abilityToRememberNewThings BOOLEAN DEFAULT false,
    levelOfKnowledge          INTEGER                                  NOT
NULL

);

CREATE TABLE IF NOT EXISTS Oddities
(
    odditiesId   SERIAL PRIMARY KEY,
    odditiesType VARCHAR(255)  NOT NULL,
    description  VARCHAR(1000) NOT NULL
);

CREATE TABLE IF NOT EXISTS MindOddities
(
    mindId     INTEGER NOT NULL,
    odditiesId INTEGER NOT NULL,
    CONSTRAINT mindId_fk FOREIGN KEY (mindId) REFERENCES Mind (mindId) ON
DELETE CASCADE,
    CONSTRAINT odditiesId_fk FOREIGN KEY (odditiesId) REFERENCES Oddities
(odditiesId) ON DELETE CASCADE,
    CONSTRAINT mindOdditiesId PRIMARY KEY (
                                          mindId, odditiesId
        )
);


CREATE TABLE IF NOT EXISTS Food
(
    foodId       SERIAL PRIMARY KEY,
    storageId    INTEGER REFERENCES Storage ON DELETE CASCADE NOT NULL,
    foodType     VARCHAR(255)                                 NOT NULL,
    numberOfFood smallint                                     NOT NULL
        CHECK (numberOfFood > 0)
);

CREATE TABLE IF NOT EXISTS Weapon
(
    weaponId       SERIAL PRIMARY KEY,
    storageId      INTEGER REFERENCES Storage ON DELETE CASCADE NOT NULL,
    weaponType     VARCHAR(255)                                  NOT NULL,
    numberOfWeapon smallint                                      NOT NULL
        CHECK (numberOfWeapon > 0)
);

CREATE TABLE IF NOT EXISTS Drugs
(
    drugId        SERIAL PRIMARY KEY,
    storageId     INTEGER REFERENCES Storage ON DELETE CASCADE NOT NULL,
    drugType      VARCHAR(255)                                 NOT NULL,
    numberOfDrugs smallint                                     NOT NULL
        CHECK (numberOfDrugs > 0)
);
CREATE TABLE IF NOT EXISTS PathHuman
(
    pathId  INTEGER NOT NULL,
    humanId INTEGER NOT NULL,
    CONSTRAINT pathId_fk FOREIGN KEY (pathId) REFERENCES Path (pathId) ON
DELETE CASCADE,
    CONSTRAINT humanId_fk FOREIGN KEY (humanId) REFERENCES Human (humanId) ON
```

```
DELETE CASCADE,
    CONSTRAINT pathHumanId PRIMARY KEY (
                                        pathId, humanId
        )

);
```

Заполнение данными:

```
INSERT INTO City (cityName, destructionStatus, coordinateX, coordinateY)
VALUES ('Диаспар', false, 5, 10);
INSERT INTO City (cityName, destructionStatus, coordinateX, coordinateY)
VALUES ('Москва', false, 100, 200);
INSERT INTO City (cityName, destructionStatus, coordinateX, coordinateY)
VALUES ('Атлантида', true, 345, 567);
INSERT INTO City (cityName, destructionStatus, coordinateX, coordinateY)
VALUES ('Афины', true, 900, 333);


INSERT INTO Human (humanName, profession)
VALUES ('Хедрон', 'Путешественник');
INSERT INTO Human (humanName, profession)
VALUES ('Олвин', 'Военный');
INSERT INTO Human (humanName, profession)
VALUES ('Иван', 'Плотник');
INSERT INTO Human (humanName, profession)
VALUES ('Александр', 'Работник склада');

INSERT INTO Network (networkName, reputation)
VALUES ('MainNetwork', 87);
INSERT INTO Network (networkName, reputation)
VALUES ('Network', 66);

INSERT INTO CityNetwork (cityId, networkId)
SELECT 1, 1
WHERE
    NOT EXISTS (
            SELECT cityId, networkId FROM CityNetwork WHERE cityId = 1 AND
networkId = 1
        );
INSERT INTO CityNetwork (cityId, networkId)
SELECT 2, 1
WHERE
    NOT EXISTS (
            SELECT cityId, networkId FROM CityNetwork WHERE cityId = 2 AND
networkId = 1
        );
INSERT INTO CityNetwork (cityId, networkId)
SELECT 3, 2
WHERE
    NOT EXISTS (
            SELECT cityId, networkId FROM CityNetwork WHERE cityId = 3 AND
networkId = 2
        );
INSERT INTO CityNetwork (cityId, networkId)
SELECT 4, 1
WHERE
    NOT EXISTS (
            SELECT cityId, networkId FROM CityNetwork WHERE cityId = 4 AND
networkId = 1
        );
```

```sql
INSERT INTO Storage (networkId, storageName, capacity)
VALUES (1, 'Склад №1', 500);
INSERT INTO Storage (networkId, storageName, capacity)
VALUES (1, 'Склад №2', 5100);
INSERT INTO Storage (networkId, storageName, capacity)
VALUES (2, 'Склад №3', 1500);

INSERT INTO Path (cityId, humanId, length)
VALUES (1, 1, 100);
INSERT INTO Path (cityId, humanId, length)
VALUES (1, 2, 150);
INSERT INTO Path (cityId, humanId, length)
VALUES (3, 1, 300);
INSERT INTO Path (cityId, humanId, length)
VALUES (2, 2, 230);
INSERT INTO Path (cityId, humanId, length)
VALUES (1, 4, 450);
INSERT INTO Path (cityId, humanId, length)
VALUES (4, 3, 500);


INSERT INTO Mind (humanId, abilityToRememberNewThings, levelOfKnowledge)
values (1, true, 120);
INSERT INTO Mind (humanId, abilityToRememberNewThings, levelOfKnowledge)
values (2, true, 112);
INSERT INTO Mind (humanId, abilityToRememberNewThings, levelOfKnowledge)
values (3, false, 87);
INSERT INTO Mind (humanId, abilityToRememberNewThings, levelOfKnowledge)
values (4, true, 133);


INSERT INTO Oddities (odditiesType, description)
VALUES ('Шизофрения', 'Возможен бред');
INSERT INTO Oddities (odditiesType, description)
VALUES ('Уход в себя', 'Может надолго уйти в раздумья');
INSERT INTO Oddities (odditiesType, description)
VALUES ('Нет странностей', 'Полностью здоров');
INSERT INTO Oddities (odditiesType, description)
VALUES ('Суицидальные мысли', 'Бывают выбросы злости и желание умереть');

INSERT INTO MindOddities (mindId, odditiesId)
SELECT 1, 1
WHERE
    NOT EXISTS (
            SELECT mindid, odditiesid FROM MindOddities WHERE mindid = 1 AND
odditiesid = 1
        );
INSERT INTO MindOddities (mindId, odditiesId)
SELECT 2, 3
WHERE
    NOT EXISTS (
            SELECT mindid, odditiesid FROM MindOddities WHERE mindid = 2 AND
odditiesid = 3
        );
INSERT INTO MindOddities (mindId, odditiesId)
SELECT 3, 2
WHERE
    NOT EXISTS (
            SELECT mindid, odditiesid FROM MindOddities WHERE mindid = 3 AND
odditiesid = 2
        );
INSERT INTO MindOddities (mindId, odditiesId)
SELECT 4, 4
WHERE
```

```
    NOT EXISTS (
            SELECT mindid, odditiesid FROM MindOddities WHERE mindid = 4 AND
odditiesid = 4
        );

INSERT INTO Food (storageId, foodType, numberOfFood)
VALUES (1, 'Рис', 200);
INSERT INTO Food (storageId, foodType, numberOfFood)
VALUES (2, 'Свинина', 100);
INSERT INTO Food (storageId, foodType, numberOfFood)
VALUES (3, 'Яблоки', 50);
INSERT INTO Food (storageId, foodType, numberOfFood)
VALUES (2, 'Хлеб', 20);
INSERT INTO Food (storageId, foodType, numberOfFood)
VALUES (3, 'Курица', 500);

INSERT INTO Weapon (storageId, weaponType, numberOfWeapon)
VALUES (1, 'AK47', 250);
INSERT INTO Weapon (storageId, weaponType, numberOfWeapon)
VALUES (3, 'T34', 10);
INSERT INTO Weapon (storageId, weaponType, numberOfWeapon)
VALUES (2, 'c300', 35);
INSERT INTO Weapon (storageId, weaponType, numberOfWeapon)
VALUES (1, 'T34', 15);


INSERT INTO Drugs (storageId, drugType, numberOfDrugs)
VALUES (2, 'Аспирин', 500);
INSERT INTO Drugs (storageId, drugType, numberOfDrugs)
VALUES (3, 'Ношпа', 300);
INSERT INTO Drugs (storageId, drugType, numberOfDrugs)
VALUES (1, 'Корвалол', 250);

INSERT INTO PathHuman (pathId, humanId)
SELECT 1, 1
WHERE
    NOT EXISTS (
            SELECT pathId, humanId FROM PathHuman WHERE pathId = 1 AND
humanId = 1
        );
INSERT INTO PathHuman (pathId, humanId)
SELECT 5, 4
WHERE
    NOT EXISTS (
            SELECT pathId, humanId FROM PathHuman WHERE pathId = 5 AND
humanId = 4
        );
INSERT INTO PathHuman (pathId, humanId)
SELECT 2, 2
WHERE
    NOT EXISTS (
            SELECT pathId, humanId FROM PathHuman WHERE pathId = 2 AND
humanId = 2
        );
INSERT INTO PathHuman (pathId, humanId)
SELECT 3, 1
WHERE
    NOT EXISTS (
            SELECT pathId, humanId FROM PathHuman WHERE pathId = 3 AND
humanId = 1
        );
INSERT INTO PathHuman (pathId, humanId)
SELECT 4, 2
WHERE
```

```
    NOT EXISTS (
            SELECT pathId, humanId FROM PathHuman WHERE pathId = 4 AND
humanId = 2
        );
INSERT INTO PathHuman (pathId, humanId)
SELECT 6, 3
WHERE
    NOT EXISTS (
            SELECT pathId, humanId FROM PathHuman WHERE pathId = 6 AND
humanId = 3
        );
```

Выводы по работе:

В результате выполнения лабораторной работы были созданы инфологическая и даталогическая модели. Получены навыки написания DDL и DML запросов на языке SQL для базы данных PostgreSQL. Некие сложности возникли из-за неинформативного текста варианта.